Remarks

The above Amendments and these Remarks are in reply to the Office Action mailed May 25, 2010, and the Advisory Action mailed August 13, 2010.

In the Advisory Action mailed August 13, 2010, it was indicated that the amendment filed by Applicant on July 26, 2010, would not be entered since it raised new issues that would require further consideration and/or search. Accordingly, Applicant respectfully requests that the amendment filed by Applicant on July 26, 2010, not be entered, and that the amendment currently enclosed herewith be entered instead.

I. Summary of Examiner's Rejections

Prior to the Office Action mailed May 25, 2010, Claims 12, 14-15, 17-20, 22, 24, 26-29, 31-34 were pending in the Application. In the Office Action, Claim 34 was rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claims 12, 14-15, 17-20, 22, 24, 26-29 and 31-32 were rejected under 35 U.S.C. 103 as being unpatentable over Connor (U.S. Patent No. 6,865,549 hereinafter Connor) in view of Huston et al. (U.S. Publication No. 2004/0093602 A1 hereinafter Huston) further in view of McLaughlin Jr. (U.S. Patent No. 7,206,805 B1 hereinafter McLaughlin), and further in view of Orton et al. (U.S. Patent No. 5,465,363 hereinafter Orton). Claim 33 was rejected under 35 U.S.C. 103 as being unpatentable over Connor in view of Huston further in view of McLaughlin and further in view of Orton as applied to Claim 22 above, and further in view of Kuftedjian (U.S. Patent No. 6,105,057 hereinafter Kuftedjian). Claim 34 was rejected under 35 U.S.C. 103 as being unpatentable over Connor in view of Freund (U.S. Patent No. 5,095,421 hereinafter Freund) and further in view of McLaughlin.

II. Summary of Applicant's Amendment

The present Reply amends Claims 12, 14-15, 22, 24, 31, 34, and adds new Claim 35, leaving for the Examiner's present consideration Claims 12, 14-15, 17-20, 22, 24, 26-29, 31-35.

III. Claim Rejections under 35 U.S.C. § 112

In the Office Action, Claim 34 was rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Applicant respectfully submits that the claim has been amended to comply with the statutory requirement under 35 U.S.C. 112. Accordingly, reconsideration thereof is respectfully requested.

IV. Claim Rejections under 35 U.S.C. § 103(a)

Claims 12, 14-15, 17-20, 22, 24, 26-29 and 31-32 were rejected under 35 U.S.C. 103 as being unpatentable over Connor in view of Huston, further in view of McLaughlin, and further in view of Orton. Claim 33 was rejected under 35 U.S.C. 103 as being unpatentable over Connor in view of Huston further in view of McLaughlin and further in view of Orton as applied to Claim 22 above, and further in view of Kuftedjian. Claim 34 was rejected under 35 U.S.C. 103 as being unpatentable over Connor in view of Freund, and further in view of McLaughlin.

Claim 12

Claim 12 has been amended to recite the following:

> *12. (Currently Amended) A system for supporting resource enlistment synchronization, comprising:*
>
> *an application server with a plurality of threads, running on one or more processors;*
>
> *a transaction manager that manages a plurality of transactions, wherein each transaction is associated with at least one said thread, and the transaction manager operates to be associated with each of the plurality of threads;*
>
> *a plurality of resource objects, wherein each resource object is wrapped with a wrapper object of a plurality of wrapper objects, wherein the transaction manager maintains and communicates with the plurality of wrapper objects to manage resource object enlistment requests from different said threads associated with different transactions;*
>
> *wherein upon receiving a request from a thread of the plurality of threads to enlist a resource object of the plurality of resource objects in a transaction, the transaction manager*
>
> > *first checks with a wrapper object of the resource object to see if there is a lock being held on the resource object by another said thread in another said transaction,*
> >
> > *if there is a lock, then allows the thread to wait and signal the thread once the lock is freed by the another said thread in the another said transaction,*
> >
> > *if there is no lock, then grants a lock to the thread and holds the lock until an owner of the thread delists the resource object, wherein the wrapper object is used to access the resource object for the thread.*

As shown above, Claim 12, as currently amended, includes features that allow the <u>interaction</u> between the transaction manager and the wrapper object of the resource object for different transactions associated with different thread.

In the pending Office Action, multiple references are cited by the Examiner, each of which is discussed individually in the following paragraphs. Applicant respectfully submits that, even though each cited reference touches on a part of the embodiment in Claim 12, there is no indication in the cited references of the interaction between the transaction manager and the wrapper object of the resource object for different transactions associated with different thread.

Connor discloses concurrency control for a policy based management system that controls resources in a distributed computing system (Abstract). Connor further discloses that, in order to facilitate a locking process for a lockable resource, a controller registers with controller services to receive controller ID and lease object for the resource (Figure 3, and Column 4, Lines 60-62).

Huston discloses a mechanism that associates a mutual exclusion lock with a shared data item in a multi-thread environment (Abstract). The mechanism includes a bit in a doorbell status register that is used to represent the availability of the shared data item to a thread. Huston further discloses that a thread examines the bit in the doorbell status register, and will continue to wait until it determines that the bit is set, or “the doorbell is rung”, by another thread, which indicates that another thread releases the shared data item (Paragraph [0047]).

McLaughlin discloses a system for executing distributed transactions (Abstract). McLaughlin further discloses that the server attempts to fork concurrent parallel threads (Column 73, Lines 27-30), and manages a complex compound transaction and separates the subtransactions into distinct threads (Column 17, Line 57 – Column 18, Line 10).

Orton discloses a view system which supports a mechanism to provide a multitask-safe wrapper for objects that are not multitask-safe (Abstract). Orton further discloses a multitasking wrapper object containing a plurality of non-multitasking target objects (Column 18, Lines 21-25).

Applicant respectfully submits that there is no indication in the cited references that, upon receiving a request from a thread of the plurality of threads to enlist a resource object of the plurality of resource objects in a transaction, the transaction manager first checks with a wrapper object of the resource object to see if there is a lock being held on the resource object by another thread in another transaction.

Additionally, there is no indication in the cited references that the transaction manager operates to be associated with each of the plurality of threads.

In view of the above comments, Applicant respectfully submits that Claim 12, as amended, is neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

Claims 12, 22, and 31

The comments provided above with regard to Claim 12 are herein incorporated by reference. Claims 12, 22, and 31 have been similarly amended to Claim 1 to more clearly recite the embodiments therein. Applicant respectfully submits that Claims 12, 22, and 31, as amended, are likewise neither anticipated by, nor obvious in view of the cited references, when considered alone or in combination. Reconsideration thereof is respectfully requested.

Claims 14-15, 17-20, 24, 26-29 and 32-33

Claims 14-15, 17-20, 24, 26-29 and 32-33 depend from and include all of the features of Claims 12 and 22. Claims 14-15, 17-20, 24, 26-29 and 32-33 are not addressed in detail herein. Applicant respectfully submits that these claims are allowable at least as they depend from an allowable independent claim, and further in view of the amendments to the independent claims, and the comments provided above. Reconsideration thereof is respectfully requested.

Claim 34

Claim 34 has been amended to recite the following:

> *34. (Currently Amended) A system for supporting resource enlistment synchronization, comprising:*
>
> *an application server with a plurality of threads, wherein the application server runs on one or more processors and is associated with a plurality of applications, wherein each application is associated with at least one said thread;*
>
> *at least one resource object, wherein the at least one resource object is associated with a resource connection object;*
>
> *a transaction manager that manages a plurality of transactions, wherein each transaction is associated with a said application;*
>
> *wherein the resource connection object operates to perform*
>
> *receiving a request, to access the at least one resource object that is associated with the resource connection object, from a application of the plurality of applications that runs on a thread of the plurality of threads,*
>
> *placing a call to the transaction manager and informing the transaction manager that current work performed by the at least one resource object is to be associated with a current transaction that is associated with the application,*
>
> *wherein, after receiving the call from the resource connection object, the transaction manager operates to perform*

first checking to see if there is an in-progress enlistment of the at least one resource object by another thread in another transaction,

if there is a lock,

blocking the request to enlist the resource object in the transaction and preventing different transactions from enlisting a logical connection to the at least one resource object at same time, and

initiating the at least one resource object to perform work associated with the thread and the current transaction, after the at least one resource object is delisted from another transaction that owns the lock,

if there is no lock,

enlisting the at least one resource object in the transaction and signaling the at least one resource object to begin processing the request,

receiving from the at least one resource object a delist resource method call, after a result is obtained for the request, to delist the at least one resource object from the current transaction and provides the result to the first said application, and

initiating the at least one resource object to perform work associated with another thread and another transaction.

As shown above, Claim 34, as currently amended, also includes features that allow the interaction between the transaction manager and the resource connection object for different applications and transactions associated with different thread.

Connor discloses concurrency control for a policy based management system that controls resources in a distributed computing system (Abstract). Connor further discloses that, in order to facilitate a locking process for a lockable resource, a controller registers with controller services to receive controller ID and lease object for the resource (Figure 3, and Column 4, Lines 60-62).

McLaughlin discloses a system for executing distributed transactions (Abstract). McLaughlin further discloses that the server attempts to fork concurrent parallel threads (Column 73, Lines 27-30), and manages a complex compound transaction and separates the subtransactions into distinct threads (Column 17, Line 57 – Column 18, Line 10).

Freund discloses that each participating resource is recorded on the log, maintained by the transaction manager, and is designated as corresponding to a particular transaction (Column 4, Lines 51-65).

However, Applicant respectfully submits that there is no indication in the cited references of the interaction between the transaction manager and the resource connection object associated with the resource object for different applications and transactions associated with different thread.

In view of the above comments, Applicant respectfully submits that Claim 34, as amended, is neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

V. Additional Amendments

Claim 35 has been newly added by the present Reply. Subject to the approval of the Examiner, Applicant respectfully requests that new Claim 35 be included in the Application and considered therewith.

VI. Conclusion

In view of the above amendments and remarks, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

Enclosed herewith is a Petition for Extension of time, together with the appropriate fee, extending the time to respond up to and including September 25, 2010.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: September 1, 2010

By: /Kuiran (Ted) Liu/
Kuiran (Ted) Liu
Reg. No. 60,039

Customer No. 80548
FLIESLER MEYER LLP
650 California Street, 14th Floor
San Francisco, California 94108
Telephone: (415) 362-3800
Fax: (415) 362-2928